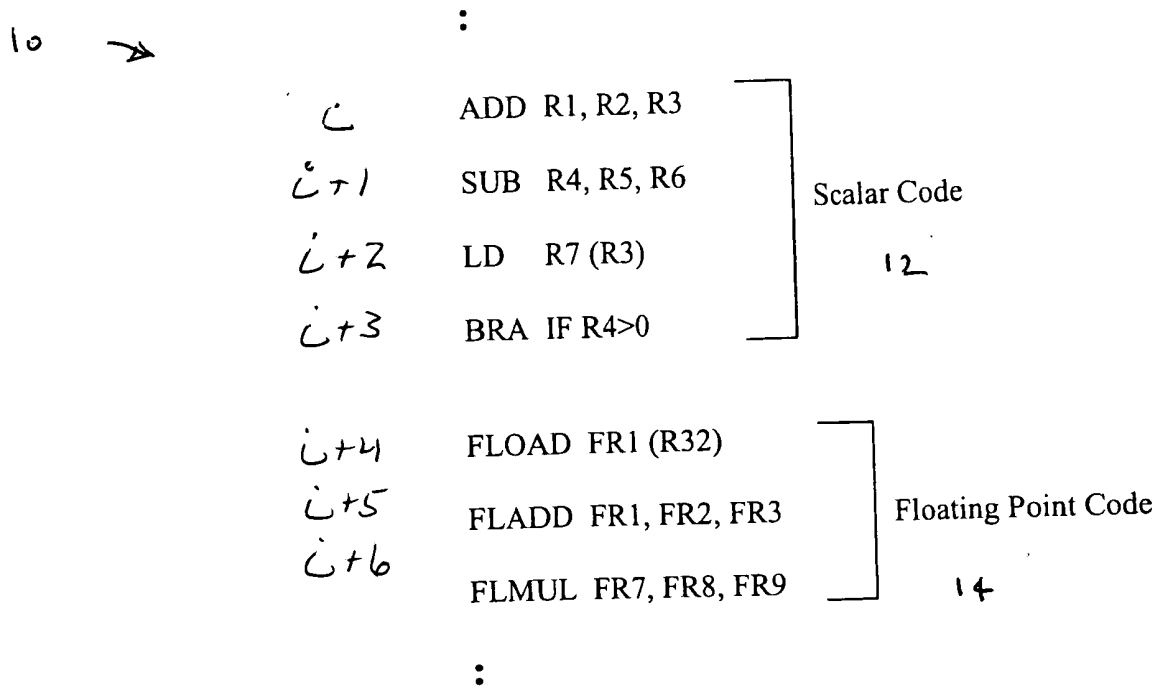
10
:
i ADD R1, R2, R3
i+1 SUB R4, R5, R6
i+2 LD R7 (R3)
i+3 BRA IF R4>0
Scalar Code
12
i+4 FLOAD FR1 (R32)
i+5 FLADD FR1, FR2, FR3
i+6 FLMUL FR7, FR8, FR9
Floating Point Code
14
:

FIG. 1

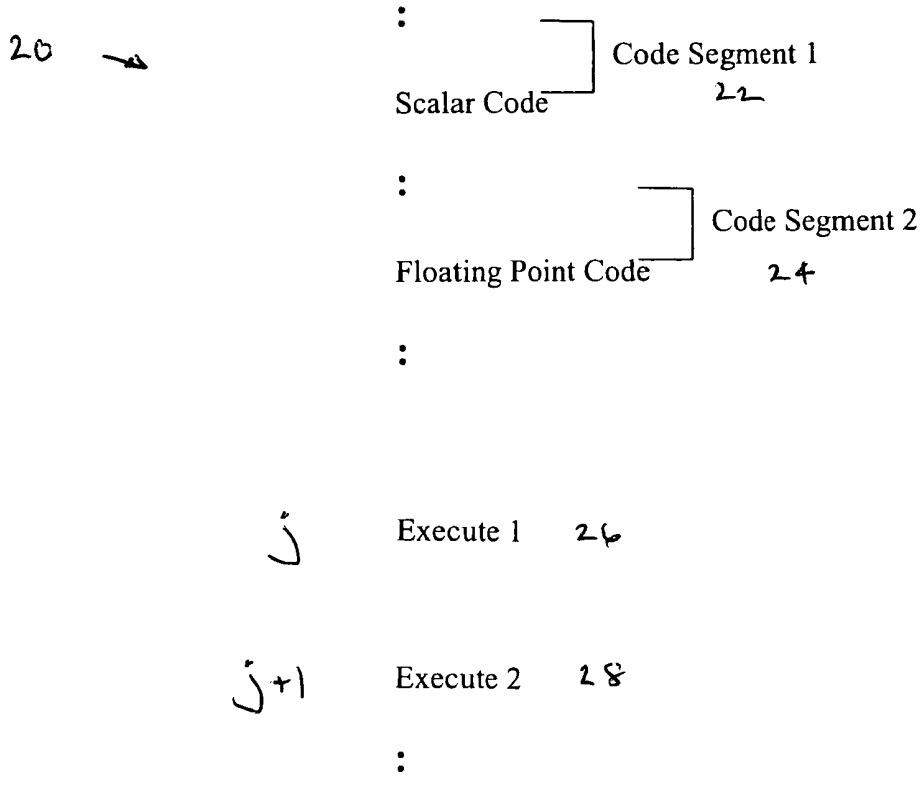
20
:
Scalar Code
Code Segment 1
22
:
Floating Point Code
Code Segment 2
24
:
j Execute 1 26
j+1 Execute 2 28
:

FIG. 2

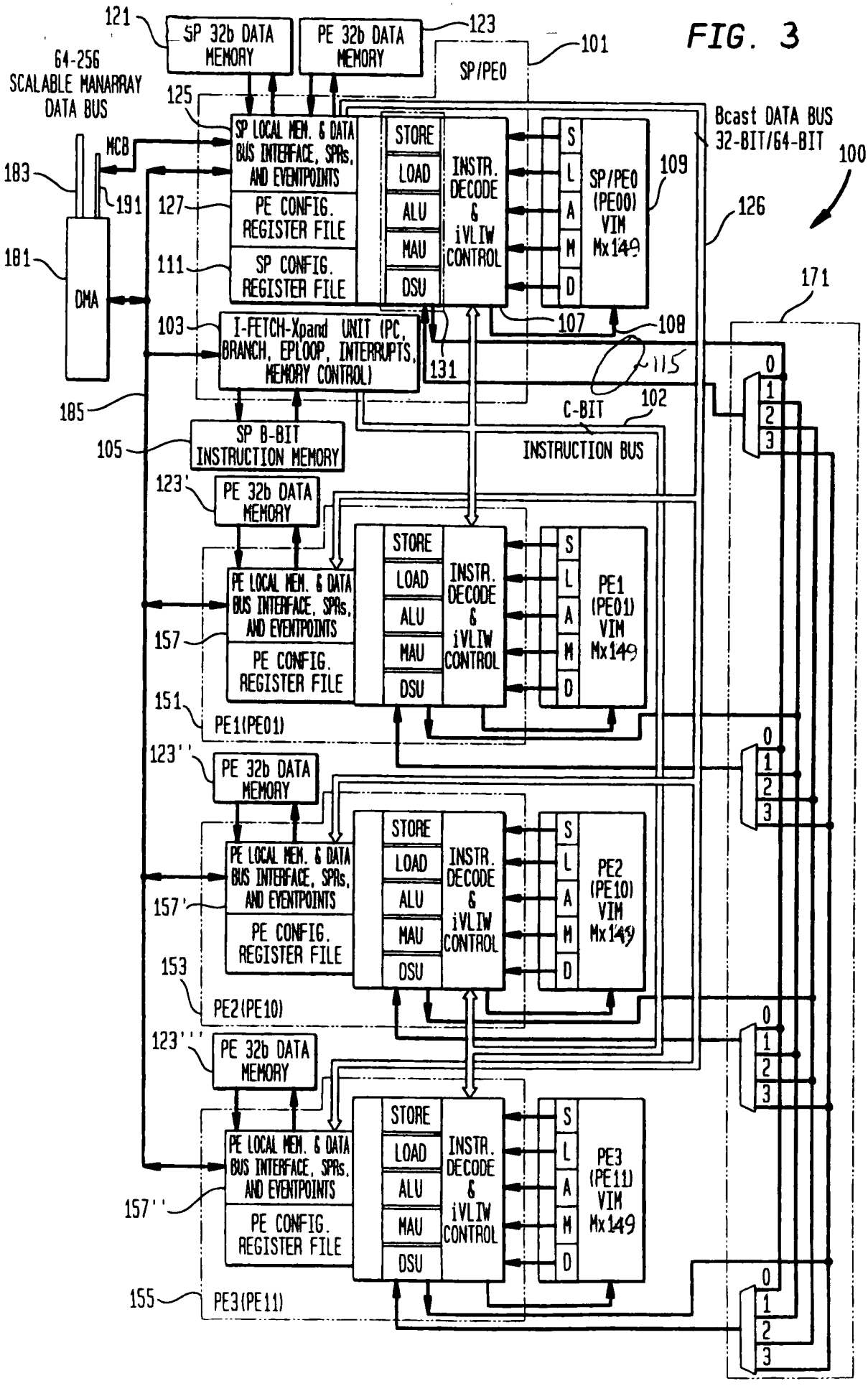
FIG. 3
121
SP 32b DATA MEMORY
PE 32b DATA MEMORY
123
101
SP/PE0
64-256 SCALABLE MANARRAY DATA BUS
125
Bcast DATA BUS 32-BIT/64-BIT
100
MCB
183
191
SP LOCAL MEM. & DATA BUS INTERFACE, SPRs, AND EVENTPOINTS
127
PE CONFIG. REGISTER FILE
111
SP CONFIG. REGISTER FILE
STORE
LOAD
ALU
MAU
DSU
INSTR. DECODE & iVLIW CONTROL
S
L
A
M
D
SP/PE0 (PE00) VIM Mx149
109
126
181
DMA
171
103
I-FETCH-Xpand UNIT (PC, BRANCH, EPLOOP, INTERRUPTS, MEMORY CONTROL)
131
107
108
115
0
1
2
3
185
C-BIT
102
105
SP B-BIT INSTRUCTION MEMORY
INSTRUCTION BUS
123'
PE 32b DATA MEMORY
PE LOCAL MEM. & DATA BUS INTERFACE, SPRs, AND EVENTPOINTS
157
PE CONFIG. REGISTER FILE
PE1 (PE01) VIM Mx149
151
PE1(PE01)
123''
PE 32b DATA MEMORY
PE LOCAL MEM. & DATA BUS INTERFACE, SPRs, AND EVENTPOINTS
157'
PE CONFIG. REGISTER FILE
PE2 (PE10) VIM Mx149
153
PE2(PE10)
123'''
PE 32b DATA MEMORY
PE LOCAL MEM. & DATA BUS INTERFACE, SPRs, AND EVENTPOINTS
157''
PE CONFIG. REGISTER FILE
PE3 (PE11) VIM Mx149
155
PE3(PE11)

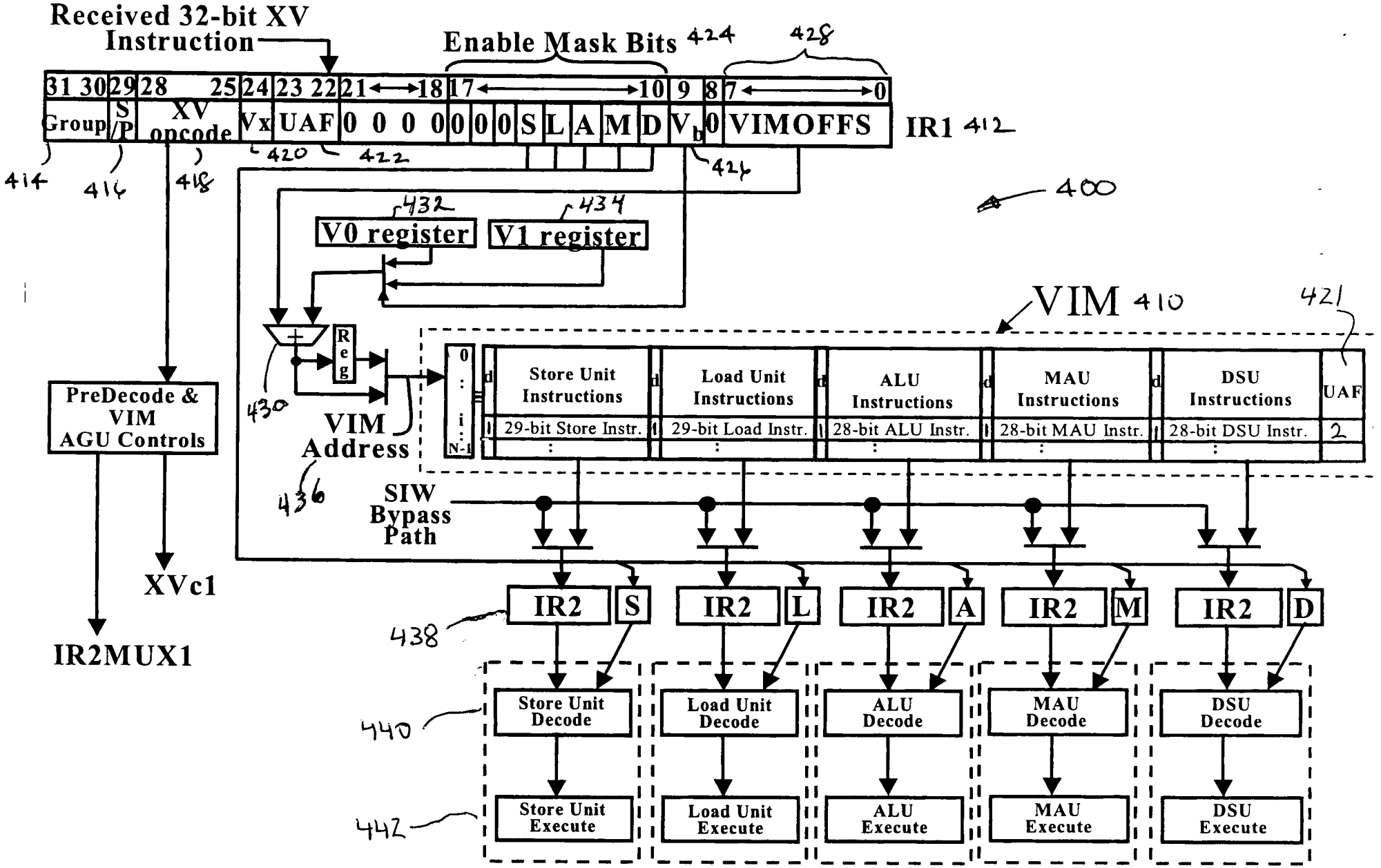
Received 32-bit XV Instruction
Enable Mask Bits 424
428
31 30 29 28 25 24 23 22 21 18 17 10 9 8 7 0
Group
S/P
XV opcode
Vx
UAF
0 0 0 0 0 0 0 S L A M D
Vb
0
VIMOFFS
IR1 412
414
416
418
420
422
424
400
V0 register
V1 register
432
434
VIM 410
421
PreDecode & VIM AGU Controls
Reg
430
VIM Address
436
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
UAF
29-bit Store Instr.
29-bit Load Instr.
28-bit ALU Instr.
28-bit MAU Instr.
28-bit DSU Instr.
2
SIW Bypass Path
XVc1
IR2MUX1
IR2 S
IR2 L
IR2 A
IR2 M
IR2 D
438
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
440
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute
442

Fig. 4

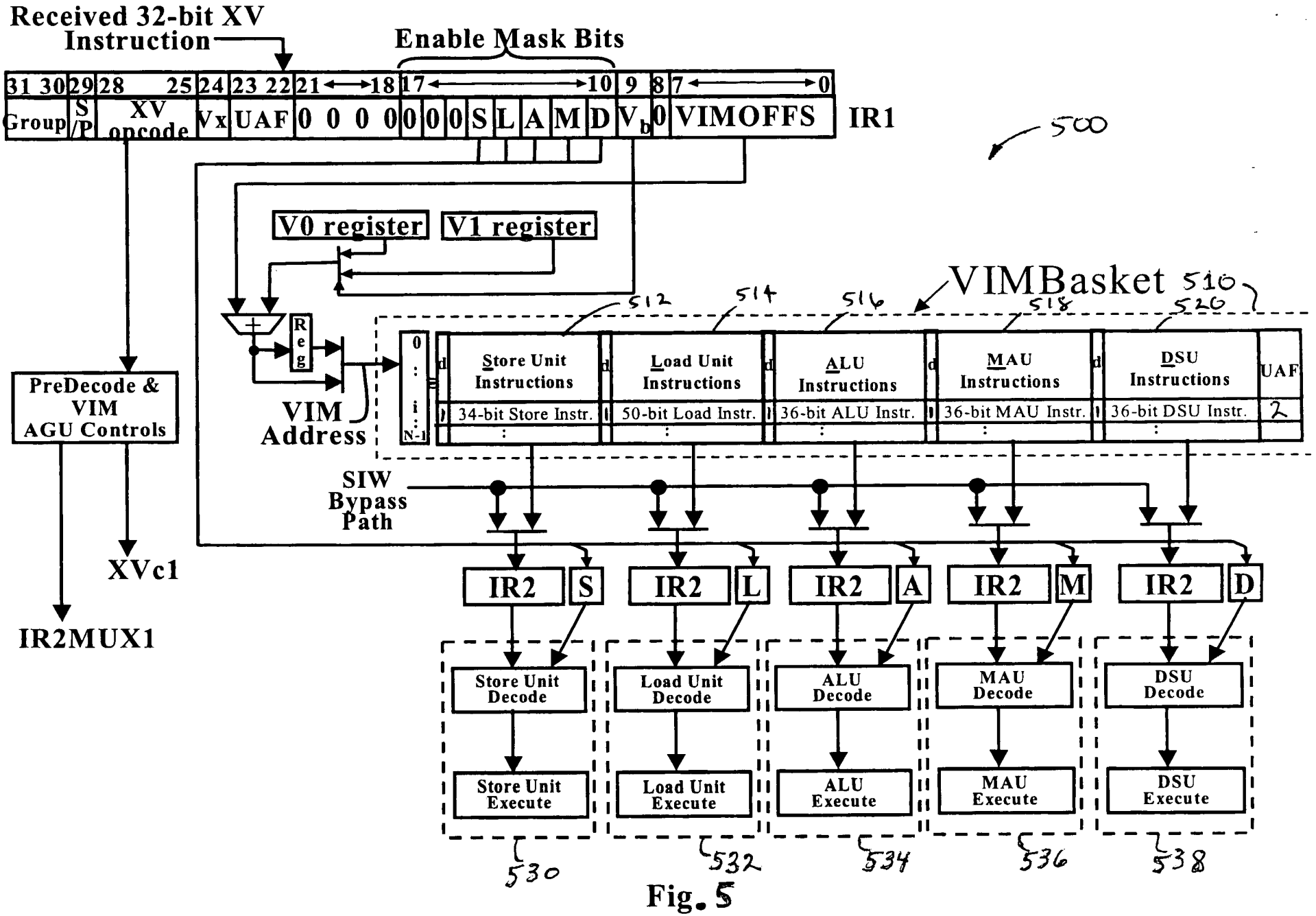
Received 32-bit XV Instruction
Enable Mask Bits
31 30 29 28 25 24 23 22 21 18 17 10 9 8 7 0
Group S/P XV opcode Vx UAF 0 0 0 0 0 0 0 S L A M D Vb 0 VIMOFFS
IR1
500
V0 register
V1 register
VIMBasket 510
512
514
516
518
520
Reg
VIM Address
0
i
N-1
d
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
UAF
34-bit Store Instr.
50-bit Load Instr.
36-bit ALU Instr.
36-bit MAU Instr.
36-bit DSU Instr.
2
PreDecode & VIM AGU Controls
SIW Bypass Path
XVc1
IR2MUX1
IR2 S
IR2 L
IR2 A
IR2 M
IR2 D
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute
530
532
534
536
538

Fig. 5

**32-bit Encoding**

600

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Group | | S/P | Unit | | MAUopcode | | | | | | Rt | | | | | Rx | | | | | Ry | | | | | CE3 | | SumpExt | | | | |
| | | | | | | | | | | | Rte | | | | 0 | Rxe | | | | 0 | Rye | | | | 0 | | | | | | | |

Fig. 6A PRIOR ART

**SLAMDunk 40-bit Encoding Example**

610

| 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Group | | S/P | Unit | | | MAUopcode' | | | | | | Rt' | | | | | | | Rx' | | | | | | | Ry' | | | | | | | | CE3 | | SumpExt' | | | | | |
| | | | | | | | | | | | | Rte' | | | | | | 0 | Rxe' | | | | | | 0 | Rye' | | | | | | 0 | | | | | | | | |

612 614

Fig. 6B

**32-bit Mapping to SLAMDunk 40-bit Encoding Example**

620

| 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Group | | S/P | Unit | | 0 | MAUopcode | | | | | | 0 | 0 | Rt | | | | | 0 | 0 | Rx | | | | | 0 | 0 | Ry | | | | | CE3 | | 0 | SumpExt | | | | |
| | | | | | | | | | | | | 0 | 0 | Rte | | | | 0 | 0 | 0 | Rxe | | | | 0 | 0 | 0 | Rye | | | | 0 | | | | | | | | |

Fig. 6C

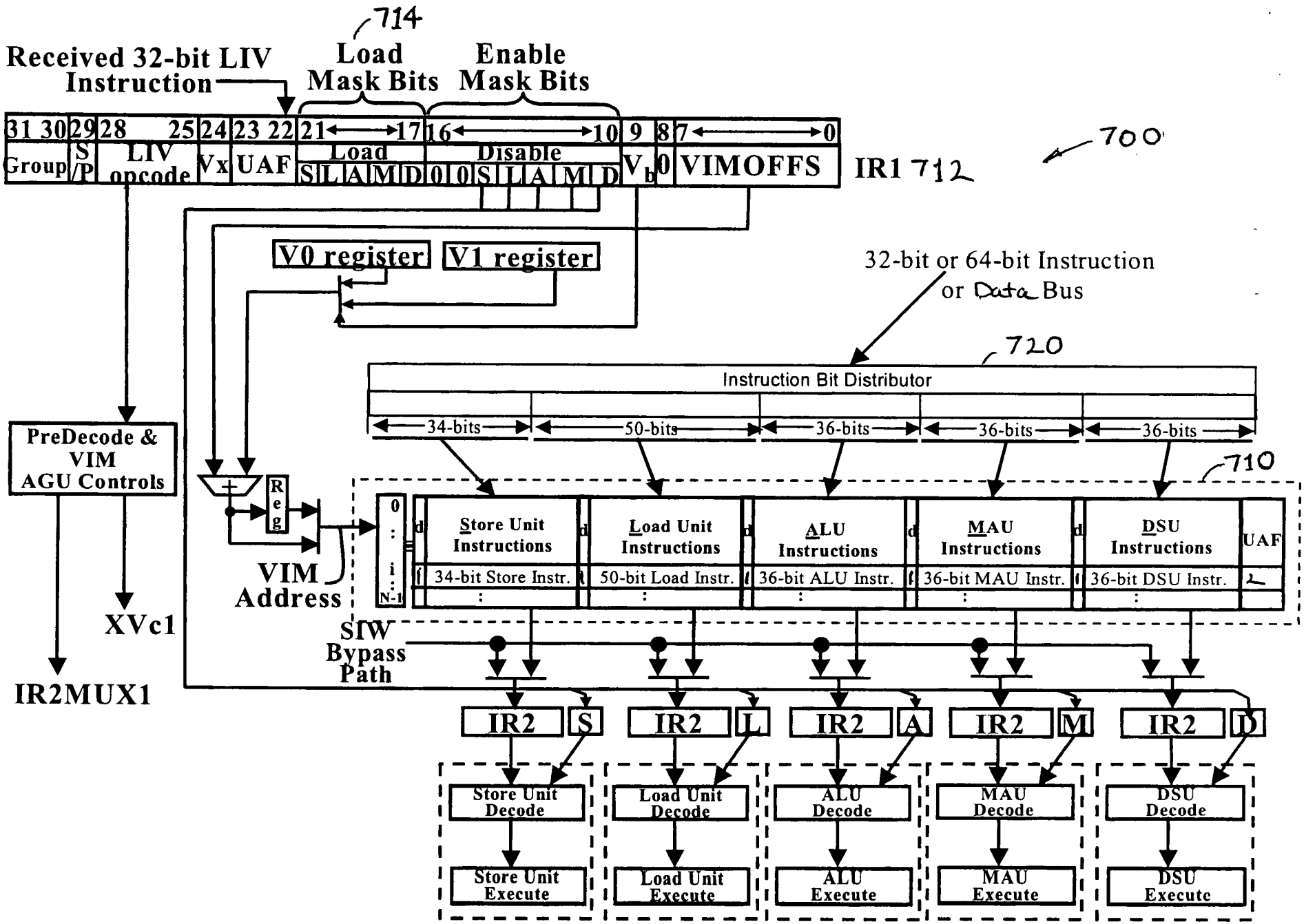
714
Received 32-bit LIV Instruction
Load Mask Bits
Enable Mask Bits
31 30 29 28 25 24 23 22 21 17 16 10 9 8 7 0
Group S/P LIV opcode Vx UAF Load S L A M D Disable 0 0 S L A M D Vb 0 VIMOFFS
IR1 712
700
V0 register
V1 register
32-bit or 64-bit Instruction or Data Bus
720
Instruction Bit Distributor
34-bits
50-bits
36-bits
36-bits
36-bits
710
PreDecode & VIM AGU Controls
Reg
VIM Address
XVc1
IR2MUX1
SIW Bypass Path
Store Unit Instructions
Load Unit Instructions
ALU Instructions
MAU Instructions
DSU Instructions
UAF
34-bit Store Instr.
50-bit Load Instr.
36-bit ALU Instr.
36-bit MAU Instr.
36-bit DSU Instr.
IR2 S
IR2 L
IR2 A
IR2 M
IR2 D
Store Unit Decode
Load Unit Decode
ALU Decode
MAU Decode
DSU Decode
Store Unit Execute
Load Unit Execute
ALU Execute
MAU Execute
DSU Execute

Fig. 7

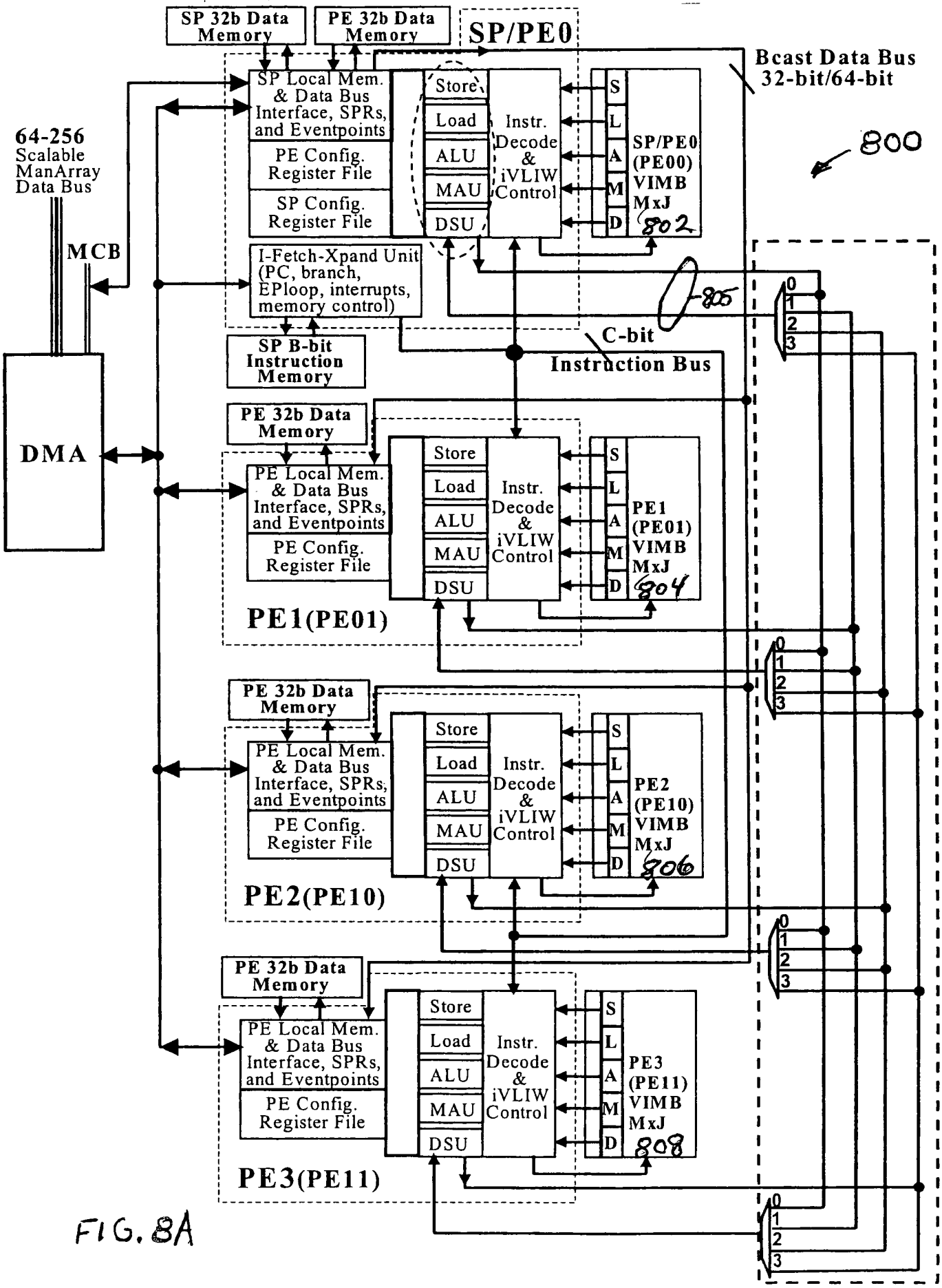
SP 32b Data Memory
PE 32b Data Memory
SP/PE0
Bcast Data Bus 32-bit/64-bit
64-256 Scalable ManArray Data Bus
MCB
800
SP Local Mem. & Data Bus Interface, SPRs, and Eventpoints
PE Config. Register File
SP Config. Register File
Store
Load
ALU
MAU
DSU
Instr. Decode & iVLIW Control
S
L
A
M
D
SP/PE0 (PE00) VIMB MxJ
802
I-Fetch-Xpand Unit (PC, branch, EPloop, interrupts, memory control)
805
SP B-bit Instruction Memory
C-bit Instruction Bus
DMA
PE Local Mem. & Data Bus Interface, SPRs, and Eventpoints
PE1 (PE01) VIMB MxJ
804
PE1(PE01)
PE2 (PE10) VIMB MxJ
806
PE2(PE10)
PE3 (PE11) VIMB MxJ
808
PE3(PE11)
0
1
2
3

FIG. 8A

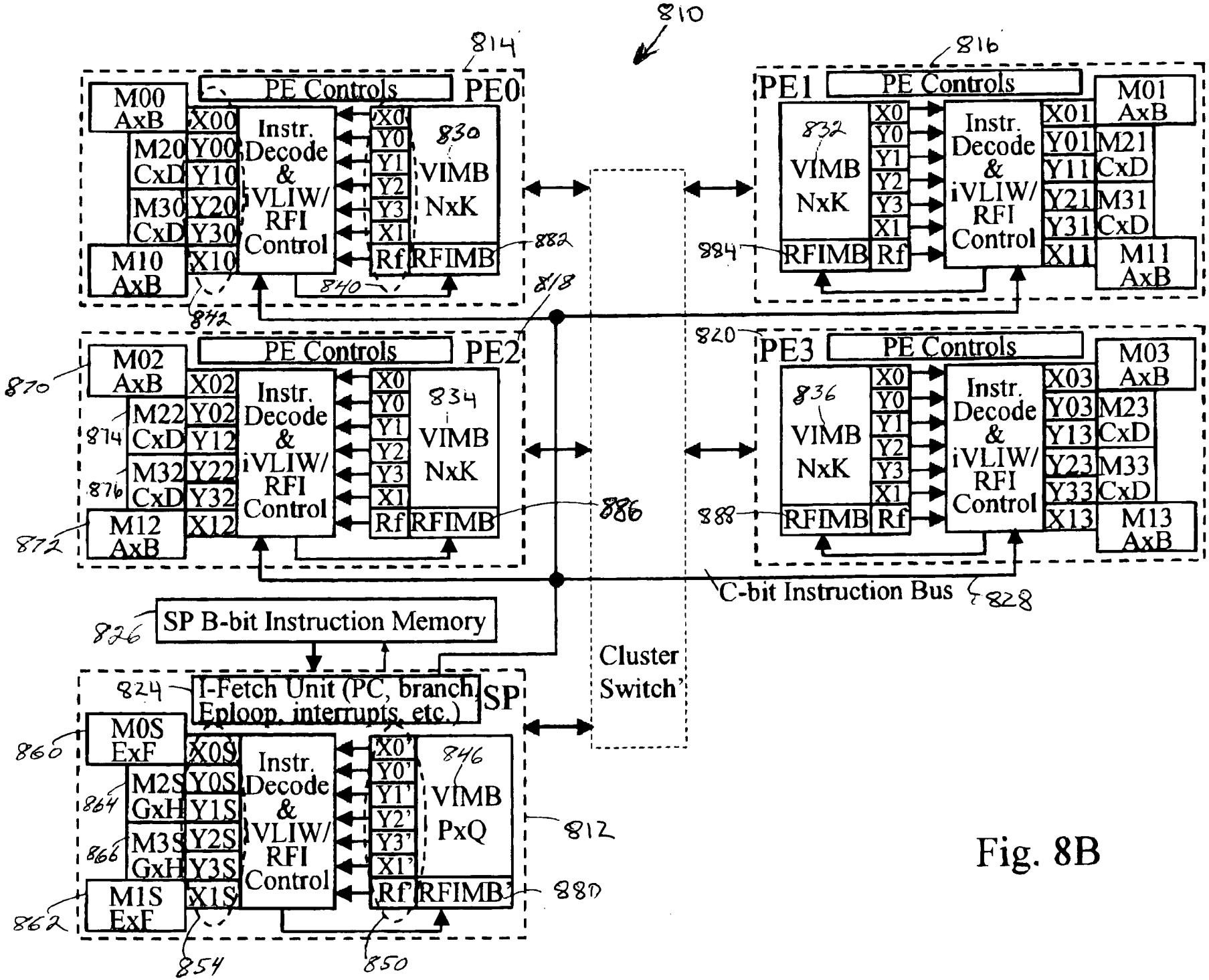
810
814
PE0
PE Controls
M00 AxB
M20 CxD
M30 CxD
M10 AxB
X00
Y00
Y10
Y20
Y30
X10
Instr. Decode & VLIW/ RFI Control
X0
Y0
Y1
Y2
Y3
X1
Rf
830
VIMB NxK
RFIMB
882
840
842
818
816
PE1
PE Controls
832
VIMB NxK
RFIMB
884
X01
Y01
Y11
Y21
Y31
X11
M01 AxB
M21 CxD
M31 CxD
M11 AxB
820
PE2
PE Controls
870
M02 AxB
874
M22 CxD
876
M32 CxD
M12 AxB
872
X02
Y02
Y12
Y22
Y32
X12
834
VIMB NxK
RFIMB
886
PE3
PE Controls
836
VIMB NxK
RFIMB
888
X03
Y03
Y13
Y23
Y33
X13
M03 AxB
M23 CxD
M33 CxD
M13 AxB
C-bit Instruction Bus
828
826
SP B-bit Instruction Memory
Cluster Switch'
824
I-Fetch Unit (PC, branch, Eploop, interrupts, etc.)
SP
860
M0S ExF
864
M2S GxH
866
M3S GxH
862
M1S ExF
X0S
Y0S
Y1S
Y2S
Y3S
X1S
X0'
Y0'
Y1'
Y2'
Y3'
X1'
Rf'
846
VIMB PxQ
RFIMB'
880
812
854
850

Fig. 8B

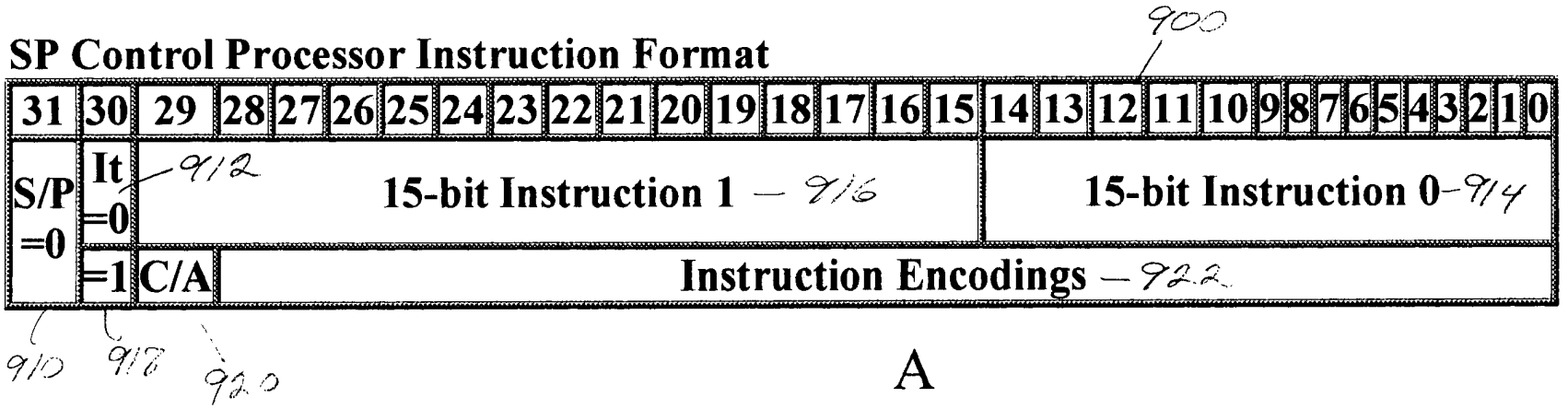
SP Control Processor Instruction Format
900
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S/P =0
It =0
912
15-bit Instruction 1 – 916
15-bit Instruction 0–914
=1
C/A
Instruction Encodings – 922
910
918
920
A
PE Instruction Format
950
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S/P =1
XV/LV/Cntl Opcode
Instruction Encodings – 964
960
962
B

Fig. 9

**XV 4/6-slot with RFI VLIW/Vector Operations – Encoding**

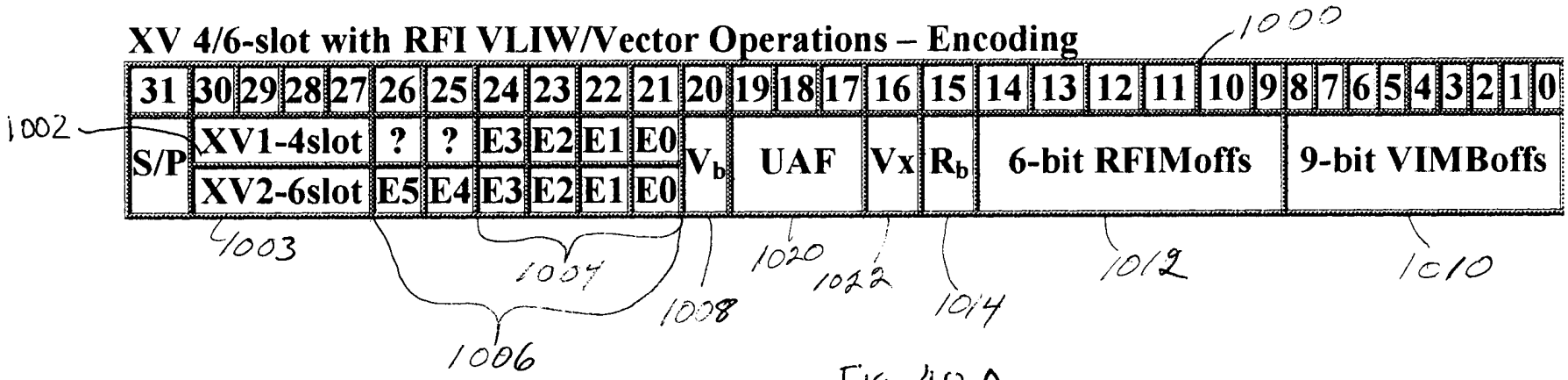


Fig. 10 A

**XV 4/6-slot with VLIW Operations – Encoding**

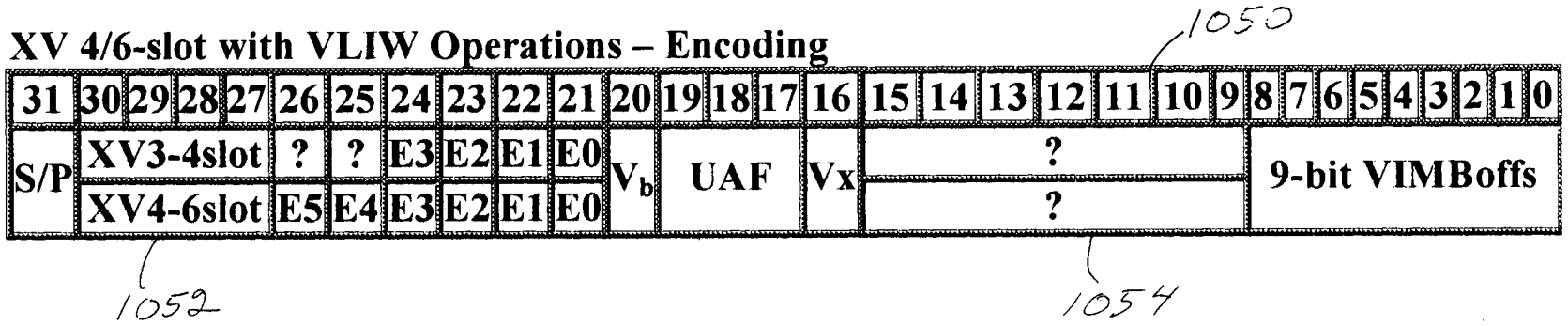


Fig. 10 B

**LV 4/6-slot Load VLIW – Encoding**

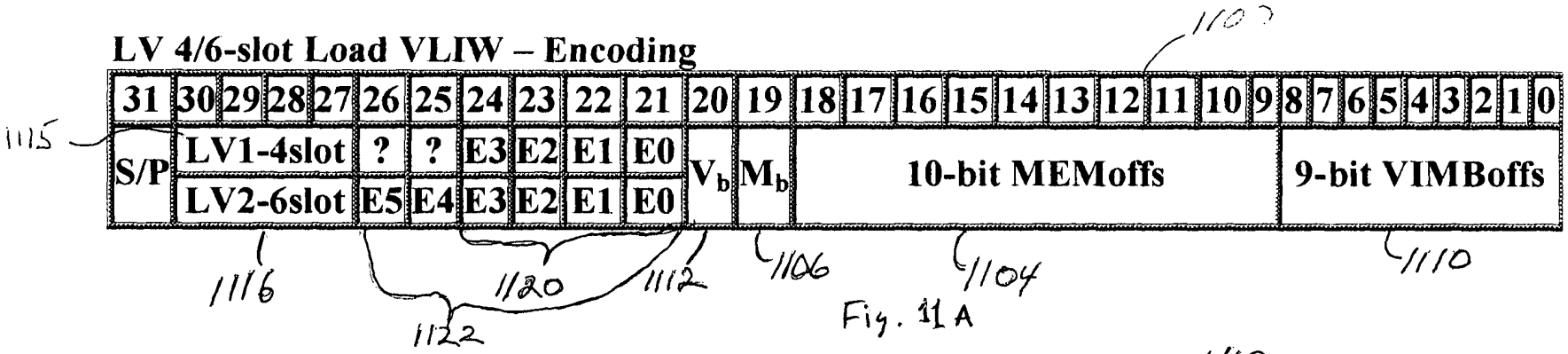


Fig. 11A

**LV 4/6-slot Load RFI Memory – Encoding**

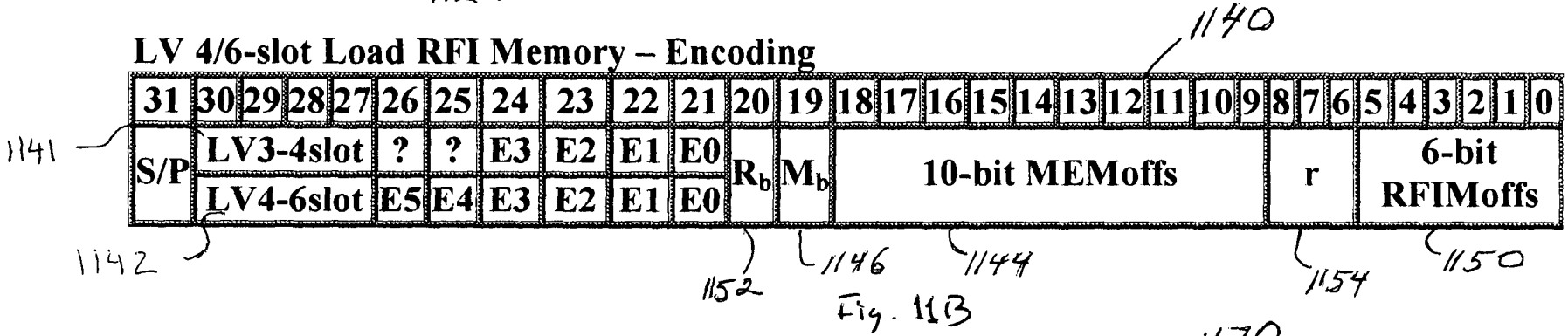


Fig. 11B

**LV 4/6-slot Load VLIW Parameters– Encoding**

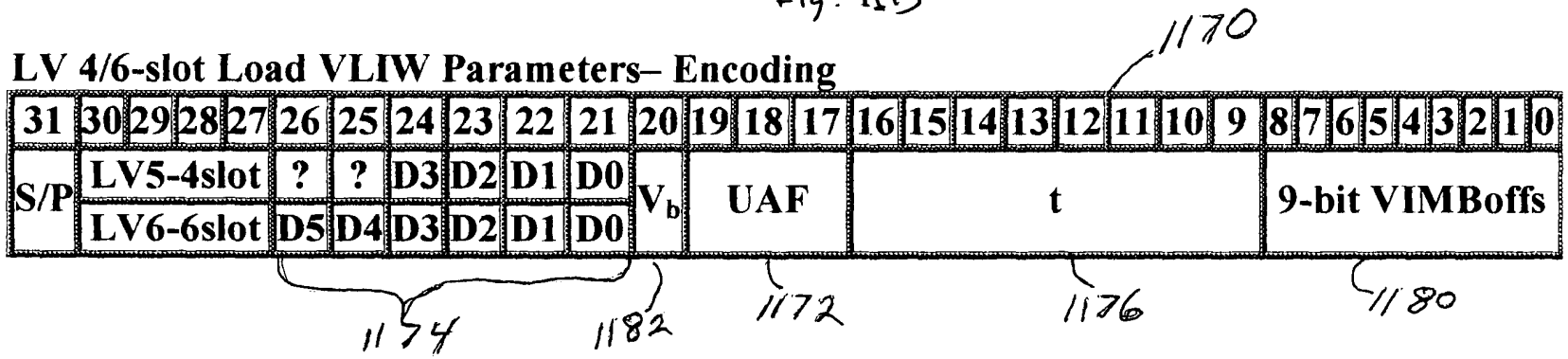


Fig. 11C

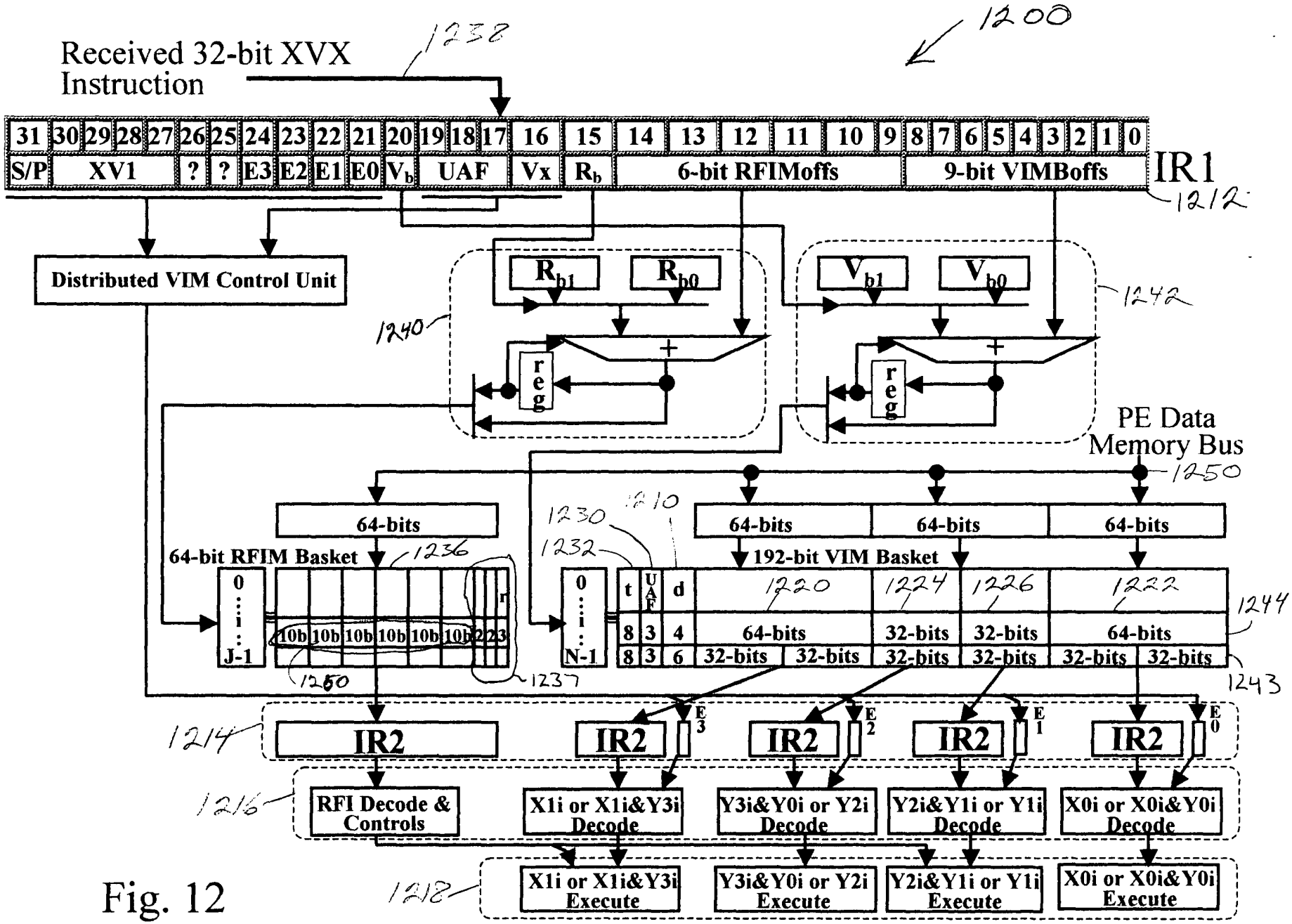
1200
Received 32-bit XVX Instruction
1238
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S/P XV1 ? ? E3 E2 E1 E0 Vb UAF Vx Rb 6-bit RFIMoffs 9-bit VIMBoffs
IR1
1212
Distributed VIM Control Unit
Rb1 Rb0
Vb1 Vb0
1240
1242
reg
reg
PE Data Memory Bus
1250
64-bits
64-bits 64-bits 64-bits
1230
1210
1232
64-bit RFIM Basket
1236
192-bit VIM Basket
0 i J-1
10b 10b 10b 10b 10b 10b 2 3
r
1250
1237
0 i N-1
t UAF d
1220 1224 1226 1222
1244
8 3 4 64-bits 32-bits 32-bits 64-bits
8 3 6 32-bits 32-bits 32-bits 32-bits 32-bits 32-bits
1243
1214
IR2 IR2 IR2 IR2 IR2
E3 E2 E1 E0
1216
RFI Decode & Controls
X1i or X1i&Y3i Decode
Y3i&Y0i or Y2i Decode
Y2i&Y1i or Y1i Decode
X0i or X0i&Y0i Decode
1218
X1i or X1i&Y3i Execute
Y3i&Y0i or Y2i Execute
Y2i&Y1i or Y1i Execute
X0i or X0i&Y0i Execute

Fig. 12

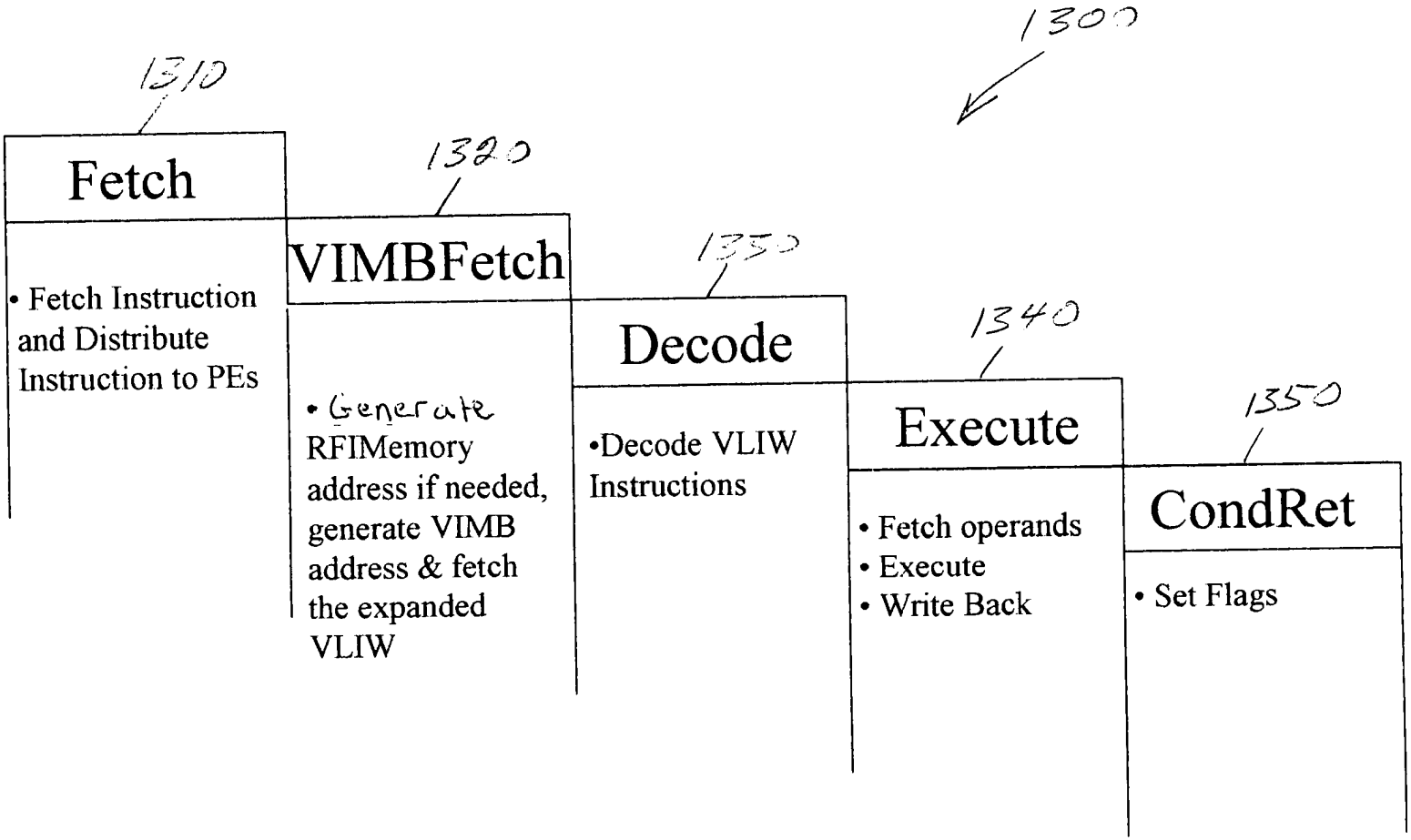
1300
1310
Fetch
• Fetch Instruction and Distribute Instruction to PEs
1320
VIMBFetch
• Generate RFIMemory address if needed, generate VIMB address & fetch the expanded VLIW
1330
Decode
•Decode VLIW Instructions
1340
Execute
• Fetch operands
• Execute
• Write Back
1350
CondRet
• Set Flags

Fig. 13

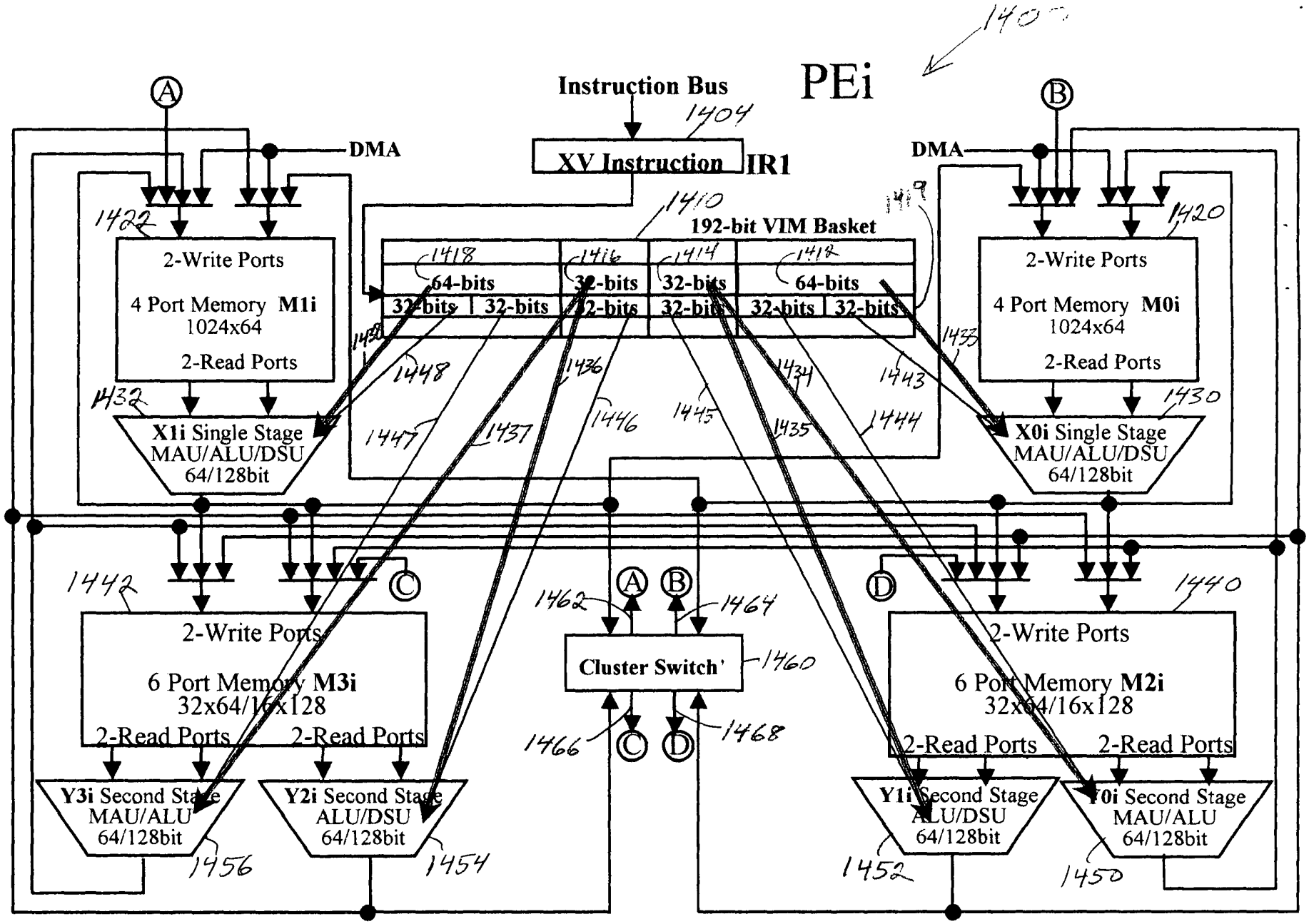
PEi
Instruction Bus
1404
XV Instruction IR1
1410
192-bit VIM Basket
DMA
DMA
1422
2-Write Ports
4 Port Memory M1i
1024x64
2-Read Ports
1432
X1i Single Stage
MAU/ALU/DSU
64/128bit
1420
2-Write Ports
4 Port Memory M0i
1024x64
2-Read Ports
1430
X0i Single Stage
MAU/ALU/DSU
64/128bit
1418
1416
1414
1412
64-bits
32-bits
32-bits
64-bits
32-bits
32-bits
32-bits
32-bits
32-bits
32-bits
1448
1436
1434
1443
1447
1437
1446
1445
1435
1444
1442
2-Write Ports
6 Port Memory M3i
32x64/16x128
2-Read Ports
2-Read Ports
1440
2-Write Ports
6 Port Memory M2i
32x64/16x128
2-Read Ports
2-Read Ports
1462
1464
Cluster Switch
1460
1466
1468
Y3i Second Stage
MAU/ALU
64/128bit
1456
Y2i Second Stage
ALU/DSU
64/128bit
1454
Y1i Second Stage
ALU/DSU
64/128bit
1452
Y0i Second Stage
MAU/ALU
64/128bit
1450

Fig. 14

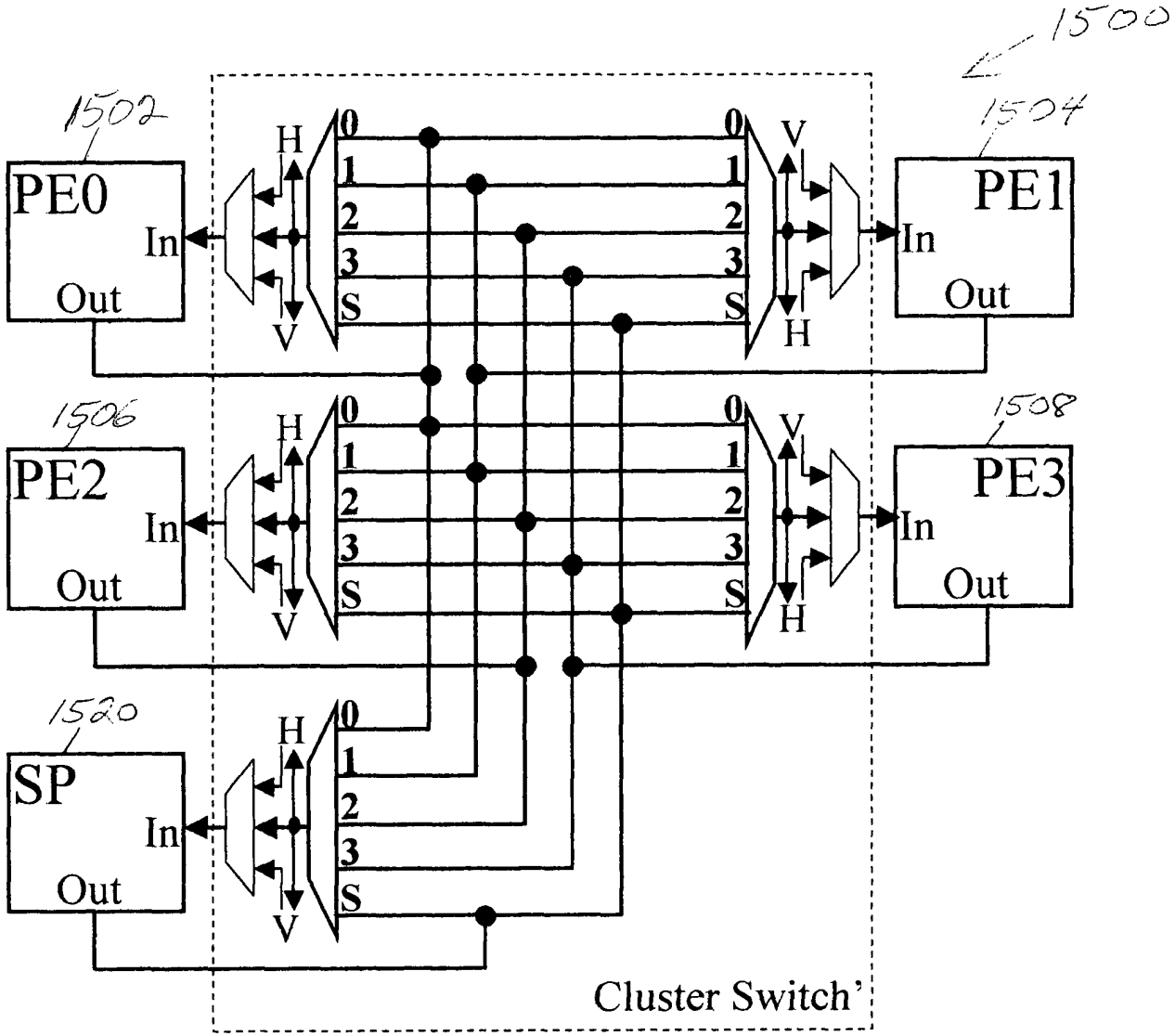
1500
1502
1504
1506
1508
1520
PE0
In
Out
PE1
In
Out
PE2
In
Out
PE3
In
Out
SP
In
Out
H
V
0
1
2
3
S
Cluster Switch'

Fig. 15

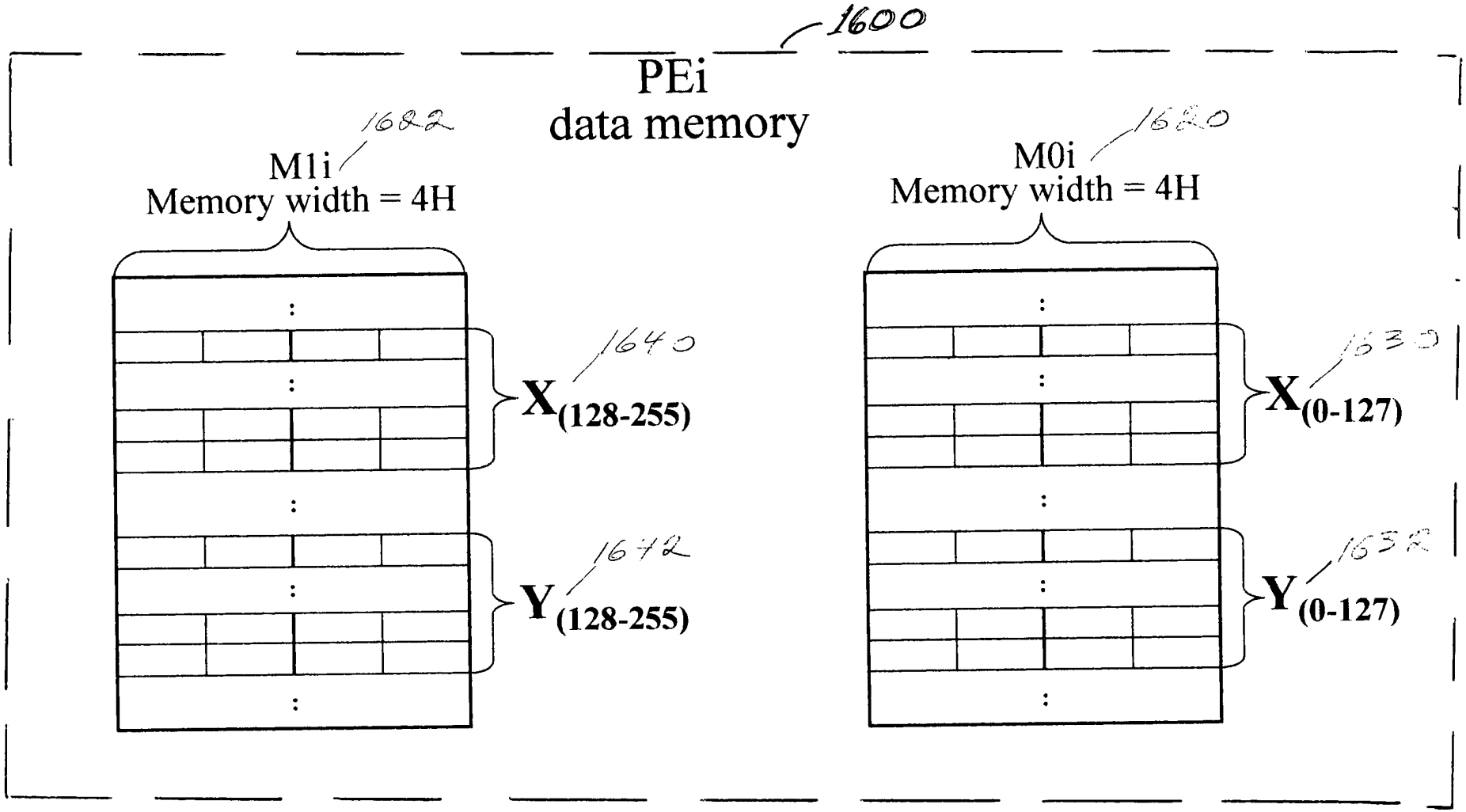
1600
PEi
data memory
1682
M1i
Memory width = 4H
1640
X(128-255)
1672
Y(128-255)
1620
M0i
Memory width = 4H
1630
X(0-127)
1652
Y(0-127)

Fig. 16

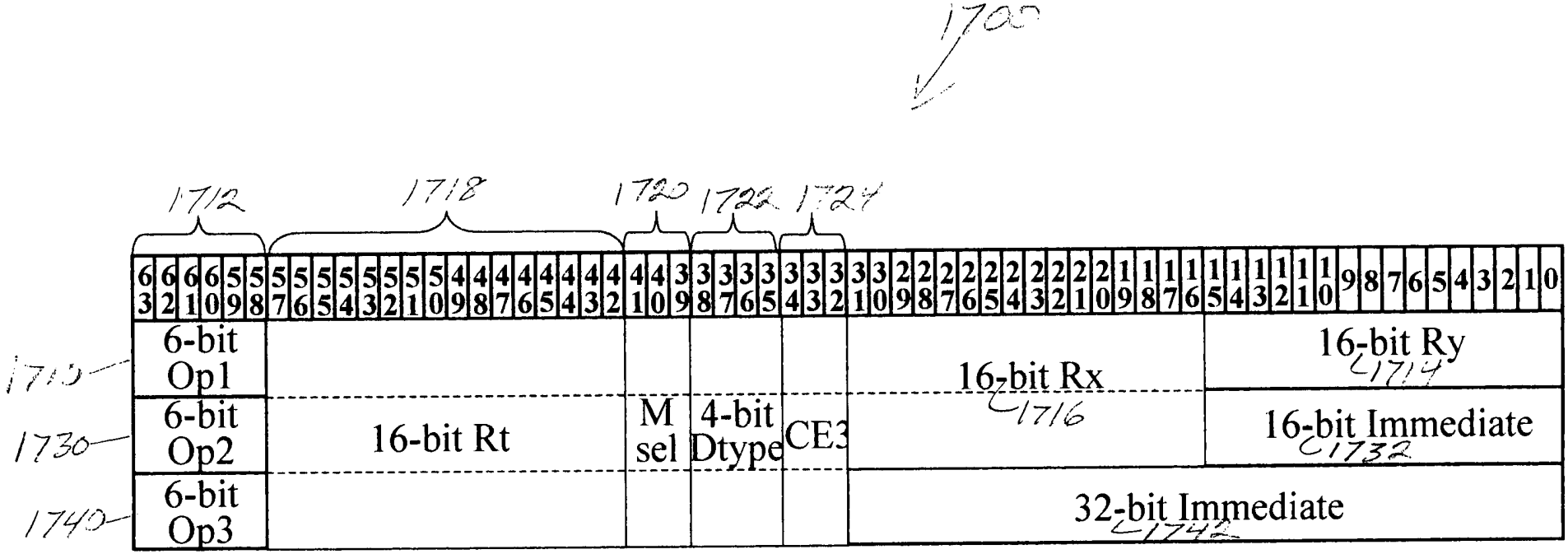
1700
1712
1718
1720
1722
1724
63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1710
6-bit Op1
16-bit Rx
1716
16-bit Ry
1714
1730
6-bit Op2
16-bit Rt
M sel
4-bit Dtype
CE3
16-bit Immediate
1732
1740
6-bit Op3
32-bit Immediate
1742

Fig. 17

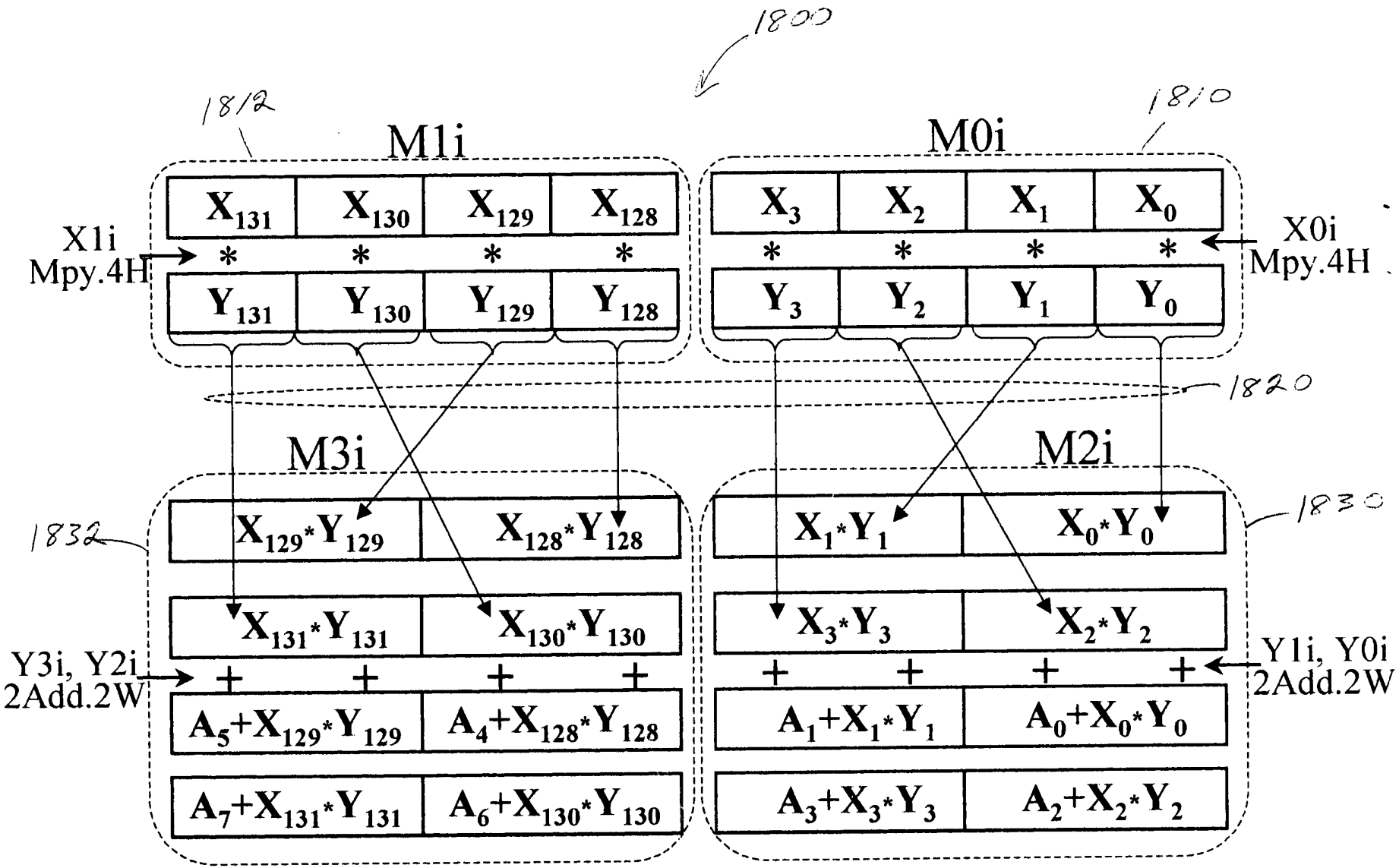
1800
1812
M1i
1810
M0i
X1i
Mpy.4H
X0i
Mpy.4H
$X_{131}$ $X_{130}$ $X_{129}$ $X_{128}$
$X_3$ $X_2$ $X_1$ $X_0$
$Y_{131}$ $Y_{130}$ $Y_{129}$ $Y_{128}$
$Y_3$ $Y_2$ $Y_1$ $Y_0$
1820
M3i
M2i
1832
1830
$X_{129}*Y_{129}$ $X_{128}*Y_{128}$
$X_1*Y_1$ $X_0*Y_0$
$X_{131}*Y_{131}$ $X_{130}*Y_{130}$
$X_3*Y_3$ $X_2*Y_2$
Y3i, Y2i
2Add.2W
Y1i, Y0i
2Add.2W
$A_5+X_{129}*Y_{129}$ $A_4+X_{128}*Y_{128}$
$A_1+X_1*Y_1$ $A_0+X_0*Y_0$
$A_7+X_{131}*Y_{131}$ $A_6+X_{130}*Y_{130}$
$A_3+X_3*Y_3$ $A_2+X_2*Y_2$

Fig. 18

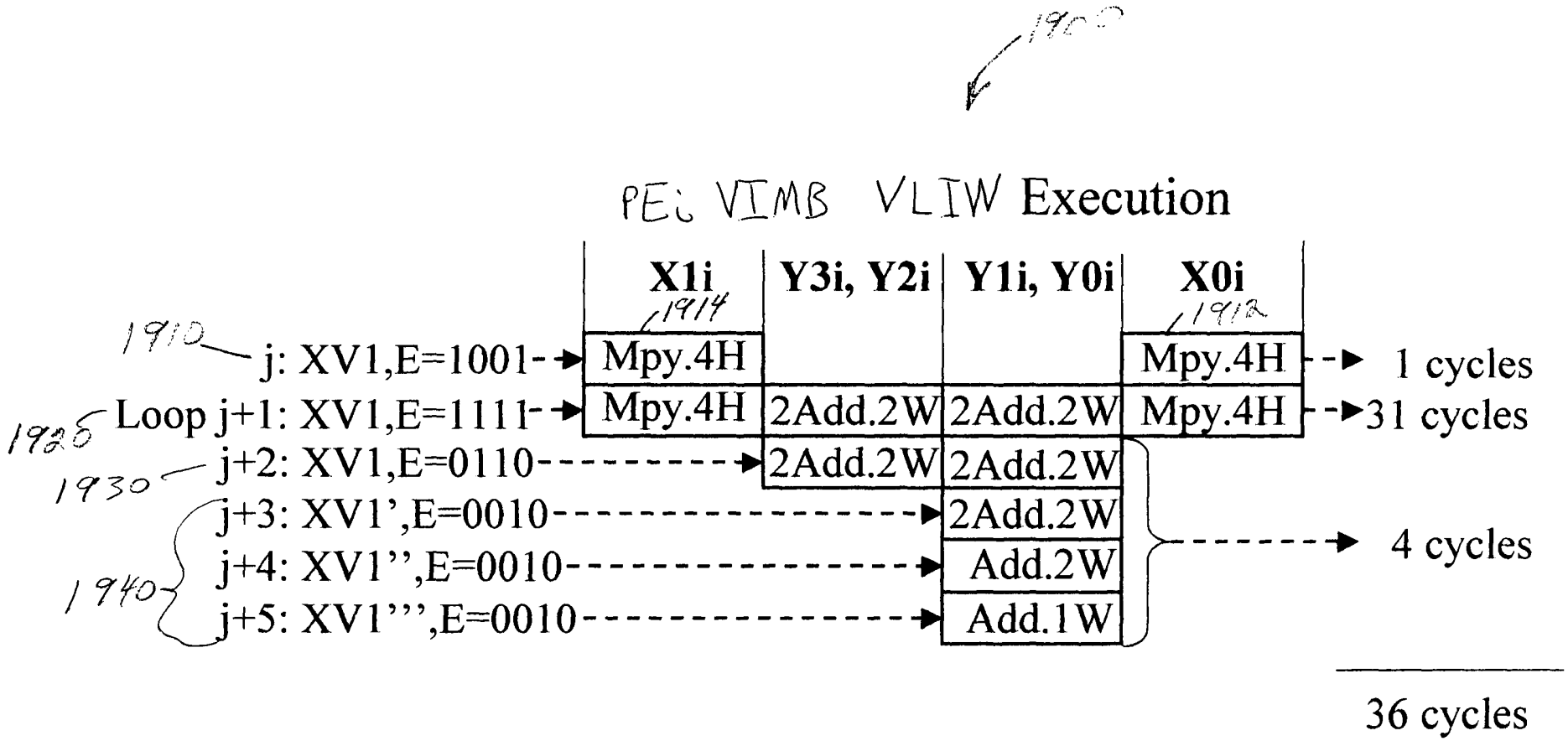
1900
PE: VIMB VLIW Execution
X1i
Y3i, Y2i
Y1i, Y0i
X0i
1914
1912
1910
j: XV1,E=1001
Mpy.4H
Mpy.4H
1 cycles
1920
Loop j+1: XV1,E=1111
Mpy.4H
2Add.2W
2Add.2W
Mpy.4H
31 cycles
1930
j+2: XV1,E=0110
2Add.2W
2Add.2W
1940
j+3: XV1',E=0010
2Add.2W
j+4: XV1'',E=0010
Add.2W
j+5: XV1''',E=0010
Add.1W
4 cycles
36 cycles

Fig. 19